Memory Interfacing in 8085

Interfacing is nothing but connecting outside peripherals to 8085 Microprocessor.

Interface is a path of communication between two components.

Interfacing is a technique of connecting microprocessor with memory.

When microprocessor is executing an instruction, it needs to access the memory for reading instruction codes and the data.

For interfacing, both the memory and the microprocessor require some signals to read from and write to registers.

Interfacing process needs to match the memory requirements and microprocessor signals.

8085 Microprocessor – 16 Address Lines (A0 – A15)

2^16 = 65536 bits = 64 KB memory.

Addresses start from 0000H to FFFFH – Addresses of the memory locations which can be addressed by the 8085 microprocessor.

8085 microprocessor uses the three control signals for the interfacing –

- IO/M’ – Input/Output or Memory
- RD’ – Read
- WR’ – Write

In response to these three signals the memory chips have also some signals –

- CE’ or CS’ – Chip enable or Chip select signal
- OE’ or RD’ – Output Enable or Read Signal (Read Operation)
- WE’ or WR’ – Write Enable or Write Signal (Write Operation)

IO/M’	RD’	WR’		
0	0	1	-	8085 reads data from memory
0	1	0	-	8085 writes data into the memory

These three signals are combined to generate two signals MEMR’ and MEMW’.

IO/M’ + RD’ = MEMR’ (0+0=0)

IO/M’ + WR’ = MEMW’ (0+0=0)

If IO/M’=1 then MEMR’ and MEMW’ signals will be deactivated irrespective of the value of RD’ and WR’ signals.

Representation of memory chip is mxn

Where, m = number of memory location/memory size

n = number of bits in each memory location

e.g. 256X8 (256 B) memory.

m = 256 = 2^8, so there are 8 address lines

n = 8, there are 8 data lines.

Prob.: Find the length of address and data bus of 2048X512 memory chip.

Sol.: 2048X512 = (2^11)X512

So, length of address = 11 and number of data lines = 512 (Answer)

Number of memory ICs required to construct a memory.

Number of ICs required = Memory to be designed / Available capacity.

Prob.: Construct 32 KB memory using 256X4 ICs.

Sol.: Number of 256X4 ICs = [32X(2^10)X8]/[256X4]

= [32X(2^10)X8]/[(2^8)X4] = 32X2^2X2 = 32X4X2 = 256

Altogether there will be 128 rows and in each row there are two 256X4 ICs

Because in 32KB, there are 8 data line whereas in 256X4, there are 4 data lines.

Prob.: Find number of 256X8 ROM chip required to design 8KB of memory.

Sol.: 256X8 = 256 B and 256X4 = 1024 B = 1 KB

To design 1 KB requires 4 chips

so, to design 8 KB requires = 8X4 = 32 Chips.

Number of Chips = 8 KB / (256X8) = [8X(2^10)X8]/[(2^8)X8] = 8X2^2 = 8X4 = 32

- **RAM (Random Access Memory):** We can read as well as write data on this type of memory. The chip of this type has pins for both memory read and memory write signals.

- **ROM (Read Only Memory):** As the name suggests, we can only write data on this type of memory chip. The chip of this type has a pin only for memory read signal.

Problem: Interface a 1kB EPROM and a 2 kB RAM with microprocessor 8085. The address allotted to 1 kB EPROM should be 3000H to 33FFH. You can assign the address range of your choice to the 2 kB RAM.

Solution: RAM and ROM both have same pins, except for WR pin, which is present in RAM and is not there in a ROM. Let us understand the pins one by one.

- **Data pins:** Since each memory location stores eight bits, there are eight data lines D0-D7 connected to the memory chip.
- **Address pins:** The number of address pins depends on the size of the memory. In this case, a memory of size 1 kB x 8 will have 2^{10} different memory locations. Hence, it will have ten address lines A0 to A9. Similarly, the 2 kB RAM will have 2^{11} different memory locations. So, there are 11 address lines A0-A10.
- **CS pin:** When this pin is enabled, the memory chip knows that the microprocessor is talking to it and responds to it accordingly. We need to generate this signal for each of the chips according to the range of addresses assigned to them. Basically, we select a chip only when it is needed. The Chip Select (CS) pin is used for this.
- **OE pin:** When this active-low output enable pin is enabled, the memory chip can output the data into the data bus.
- **WR pin:** Upon activation of this active-low memory write pin, data on the data bus is written on the memory chip at the location specified by the address bus.
- **VCC and GND pins:** These pins serve the purpose of powering the ICs. For simplicity, we will not show these pins in the diagram.

There are three types of buses in 8085 – Address bus, data bus, and control bus. Each of these buses will be connected to the memory chip.

Let us tabulate the starting and ending address of the 1kB EPROM.

A15 is most significant, and A0 is the least significant bit. The address range for placing the EPROM is from 3000H to 33FFH (as given in the question.) Translating these to binary:

3000H = 0011 0000 0000 0000

33FFH = 0011 0011 1111 1111

Address bit number	**A15**	**A14**	**A13**	**A12**	**A11**	**A10**	**A9**	**A8**	**A7**	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
Starting address	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Ending Address	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1

From the above table, we can observe that ten bits from A0 to A9 are changing. These ten bits are directly connected to the address lines of the memory chip.

These ten bits take the value of either 0 or 1 to form addresses. The first address is 00 0000 0000, and the second address is 00 0000 0001, the third is 00 0000 0010 and so on. The last address will be 11 1111 1111.

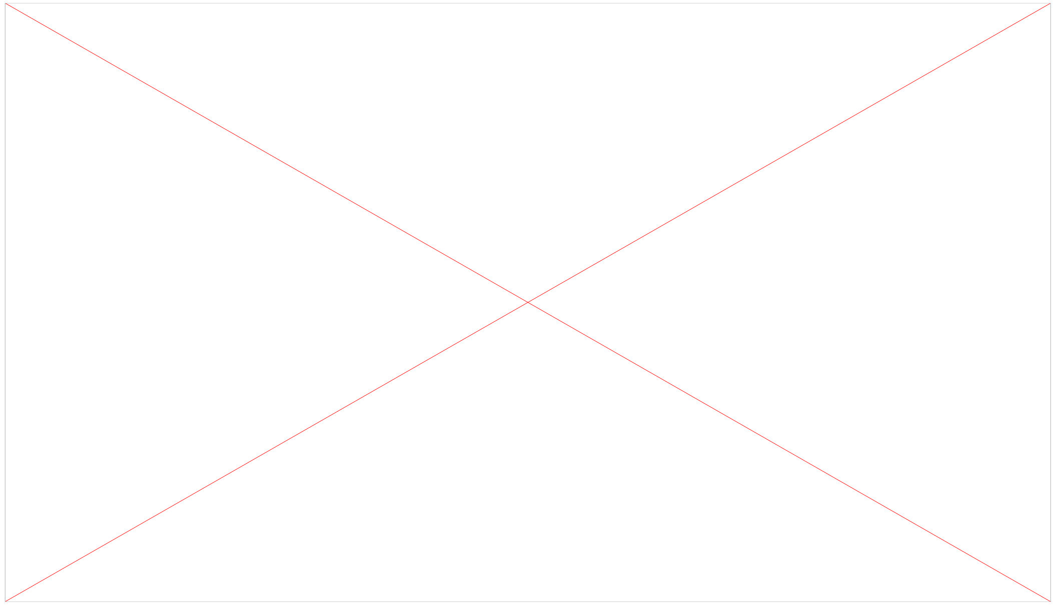
Meanwhile, bits A11 to A15 do not change and don’t have any effect on the addressing process inside the memory chip. So, we can conclude that the values of bits A15-A11 (0011 00) given in the above table are in a unique, unchanging configuration for this memory chip. If even one of these bits changes, the address won’t belong to this memory chip. **So, we can use these values of A15-A11 to uniquely identify this memory chip, which is exactly what the CS signal is supposed to do.**

We can say that when A15 = A14 = A11 = A10 = 0 and A13 = A12 = 1, then our memory chip should be selected.

Now, we need to design the logic to generate the CS signal. The resulting Boolean equation of CS will be:

CS = Complement of (A15* . A14* . A13 . A12 . A11* . A10*)

This equation can be implemented using NAND Gate. The final chip select logic for 1kB EPROM is illustrated below.



Now, we have to generate a chip select signal for the second memory chip, which is 2kB RAM. The process is quite similar and differs from the previous one in two ways:

- The size of the memory is different. So, there are 11 address lines instead of 10.
- We are not given an address range here. We are given the liberty to decide on our own.

Similar to the previous case, we connect the first 11 address lines of the 8085 microprocessor to the 11 address lines of the 2kB RAM. These bits will take values of 0 and 1 and will generate 2 * 1024 different addresses. The address bits A10-A0 will vary from 000 0000 0000 to 111 1111 1111.

What about the remaining address bits? Well, they don't have any role in the addressing of the memory in this 2kB RAM. So, we can fix them to a certain value without affecting anything. Let's fix them to 0000 0. Thus, the address range for this chip becomes 0000 0000 0000 0000 to 0000 0111 1111 1111. In hexadecimal, the address range will be from 0000H to 07FFH.

Address bit number	**A15**	**A14**	**A13**	**A12**	**A11**	**A10**	**A9**	**A8**	**A7**	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
Starting address	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ending Address	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

We use a similar technique here. We use the remaining bits A15-A11 to uniquely identify this chip i.e., to generate chip select signal. So, the boolean equation will be

CS = Complement of (A15* . A14* . A13* . A12* . A11*)

The implementation of this equation using NAND Gate to generate the CS signal is shown in the following image.

The final circuit

Since we now have the chip select logic and have decided all the connections, it’s time to finalize the circuit. The entire external memory interfacing circuit can be broken up into five different parts:

- 8085 microprocessor
- Demultiplexing of address/data bus
- Generation of control signals
- Generation of chip select signals
- Memory chips

The images below show the final circuit with all the five parts listed above integrated into a single circuit. Just the connections are shown in the first diagram. In the diagram following it, different subsections of the circuit are labeled.

Peripherals are connected to microprocessor by two modes –

- Memory mapped I/O mode (I/O devices are treated as memory ICs) and I/O mapped I/O mode

	Memory mapped I/O		**I/O mapped I/O**	
Number of Address lines	16	16	16	8
Control Signals	MEMR’ MEMW’	MEMR’ MEMW’	MEMR’ MEMW’	IOR’ IOW’
Number of peripherals	64 KB → Mem.+I/O (e.g. 64+0, 50+14)		64 KB	2^8 = 256 I/O devices

Memory mapped I/O:

Advantages:

- IO/M’ is not required. So no separate instructions are needed.
- Arithmetic, logical operations can be directly performed on I/O data.

Disadvantages:

- Interfacing is complex.
- Memory space required is high.

I/O mapped I/O:

Advantages:

- Interfacing is less complex.
- Maximum capacity of microprocessor will be utilized.

Disadvantages:

- I/O devices require separate instructions.
- Arithmetic, logical operations can’t be performed on I/O data.
- 4 control signals required.